

# AquesTalk1 Linux マニュアル

株式会社 アクエスト  
[www.a-quest.com](http://www.a-quest.com)

## 概要

本文書は、音声合成ライブラリ AquesTalk1 Linux をアプリケーションに組み込んで使用するためのプログラミングの方法、注意点を示したものです。

AquesTalk1 は、かな表記の音声記号列から WAV 音声データを生成するライブラリです。

声種の変更は so ライブラリを差し替えて行います。

本ライブラリを使用するには、開発ライセンスキーの設定が必要です。このライセンスキーを設定しない場合は、評価版として動作し、以下の制限があります。

### 評価版の制限

「ナ行、マ行」を指定すると、すべて「ヌ」と発声します

また、本ライブラリをアプリケーションに組み込んで使用する際には**使用ライセンス**、配布には**頒布ライセンス**が必要です。ライセンスの種類や購入方法は、弊社サイトのライセンスのページを参照してください。

## 仕様

ライブラリ形式	So 形式共有ライブラリ x86-64 / i386
対応 OS	各種 Linux
入力データ形式	かな表記音声記号列 (Shift JIS/UTF8/UTF16)
出力データ形式	WAV フォーマット (8KHz サンプリング, 16bitPCM, モノラル)
声種	9 種
関数 I/F	C 関数呼び出し
マルチスレッド	対応
ライブラリサイズ	約 150KByte

## ビルド・実行

### ヘッダ、ライブラリ

プログラムのコンパイル時にはヘッダファイル(AquesTalk.h)をインクルードします。必要に応じてインクルードファイルのパスを設定しておきます。リンク時には、`-lAquesTalk` を指定して `libAquesTalk.so` ファイルをリンクします。

声種、64/32bit 毎に `libAquesTalk.so` という同名の異なるファイルがあります。使用する声種、動作環境に応じて選択して用いてください。動的に声種を変更する場合は、`dlopen()`などで実行時に動的ロードします。

### 実行

アプリ(実行モジュール)の実行の際には、`libAquesTalk.so` が必要です。

### 依存ライブラリ

`libAquesTalk.so` は、`libc.so.6` を呼び出しています。

`AquesTalk_Synthe()`関数で返された音声データは、使用後に `AquesTalk_FreeWav()`で解放してください。アプリ側で `free()` などを使用して解放すると、異なる CRT の場合にハングします。

## 関数 API

### AquesTalk\_Synthe

AquesTalk.h

説明	かな表記音声記号列(SJIS)から音声波形を生成します
構文	<code>unsigned char * AquesTalk_Synthe(const char *koe, int speed, int *size)</code>
引数	
<i>koe</i>	音声記号列(SJIS NULL 終端)を指定
<i>speed</i>	発話速度[%] 50-300 の間で指定 デフォルト:100 値を大きく設定するほど、速くなる
<i>size</i>	生成した音声データのサイズが返る[byte](エラーの場合はエラーコードが返る)
戻り値	WAV フォーマットの音声データを返す。 ヒープ領域を関数内部で確保するので、解放は <code>AquesTalk_FreeWave()</code> で行う。 エラー時は、NULL を返す。このとき <i>size</i> にエラーコードが設定される。

### AquesTalk\_Synthe\_Utf8

AquesTalk.h

説明	かな表記音声記号列(UTF-8)から音声波形を生成します
構文	unsigned char * <b>AquesTalk_Synthe_Utf8</b> (const char *koe, int speed, int *size)
引数	
koe	音声記号列(UTF8 NULL 終端 BOM は付与しない)を指定
speed	発話速度[%] 50-300 の間で指定 デフォルト:100 値を大きく設定するほど、速くなる
size	生成した音声データのサイズが返る[byte](エラーの場合はエラーコードが返る)
戻り値	WAV フォーマットの音声データを返す。 ヒープ領域を関数内部で確保するので、解放は AquesTalk_FreeWave()で行う。 エラー時は、NULL を返す。このとき size にエラーコードが設定される。

### AquesTalk\_Synthe\_Utf16

AquesTalk.h

説明	かな表記音声記号列(UTF-16)から音声波形を生成します
構文	unsigned char * <b>AquesTalk_Synthe_Utf16</b> (const unsigned short *koe, int speed, int *size)
引数	
koe	音声記号列(UTF16 NULL 終端 BOM の付与は任意)を指定
speed	発話速度[%] 50-300 の間で指定 デフォルト:100 値を大きく設定するほど、速くなる
size	生成した音声データのサイズが返る[byte](エラーの場合はエラーコードが返る)
戻り値	WAV フォーマットの音声データを返す。 ヒープ領域を関数内部で確保するので、解放は AquesTalk_FreeWave()で行う。 エラー時は、NULL を返す。このとき size にエラーコードが設定される。

### AquesTalk\_FreeWave

AquesTalk.h

説明	音声データの領域を開放
構文	void <b>AquesTalk_FreeWave</b> (unsigned char *wav)
引数	
wav	WAV フォーマットのデータ(AquesTalk_Synthe()等で生成した音声データ)
戻り値	なし

### AquesTalk\_SetDevKey

AquesTalk.h

<b>説明</b>	開発ライセンスキーを設定。音声波形を生成する前に一度呼び出すことで、以降、製品版とし動作し、評価版の制限がなくなる。
<b>構文</b>	int <b>AquesTalk_SetDevKey</b> (const char *key)
<b>引数</b>	
key	開発ライセンスキー文字列(半角英数)
<b>戻り値</b>	ライセンスキーが正しければ 0、正しくなければ 1 が返る。 不正なキーでも 0 を返す場合がある。このとき制限は解除されない。

## AquesTalk\_SetUsrKey

AquesTalk.h

<b>説明</b>	使用ライセンスキーを設定。音声波形を生成する前に一度呼び出すことで、以降、合成音声データに含まれる透かしが使用ライセンス無しから取得済みに変化する。
<b>構文</b>	int <b>AquesTalk_SetUsrKey</b> (const char *key)
<b>引数</b>	
key	使用ライセンスキー、または頒布ライセンスキーの文字列(半角英数)
<b>戻り値</b>	ライセンスキーが正しければ 0、正しくなければ 1 が返る。 不正なキーでも、まれに 0 を返す場合がある。このときはライセンス無しのままである。

## 音声記号列

AquesTalk1 は、かな表記の音声記号列から音声を作成します。漢字を含んだテキスト文字列から音声を作成するときは、別途、言語処理ライブラリ AqKanji2Koe を用いて漢字仮名交じりテキストから音声記号列に変換する必要があります。

音声記号列の詳細は、付属の音声記号列仕様書を参照してください。

※AquesTalk1 Ver.2.0 から「フュ」などの音声記号列の音韻が拡張されています。

## エラーコード表

関数が返すエラーコードの内容は、次の通りです。

値	内容
100	その他のエラー
101	メモリ不足
105	音声記号列に未定義の読み記号が指定された

105	音声記号列に未定義の読み記号が指定された
106	音声記号列のタグの指定が正しくない
107	タグの長さが制限を越えている(または[>]が見つからない)
108	タグ内の値の指定が正しくない
200	音声記号列が長すぎる
201	1つのフレーズ中の読み記号が多すぎる
202	音声記号列が長すぎる
203	ヒープメモリ不足
204	音声記号列が長すぎる

## サンプルプログラム

次に示すコードは、標準入力から音声記号列を1行読み込み、標準出力に音声波形(.wav フォーマット)を出力するプログラムです。同じソースが、パッケージに含まれています(SampleTalk.c)。

```
#include <stdio.h>
#include <AquesTalk.h> // AquesTalk クラスのヘッダ

int main(int ac, char **av)
{
    int size;
    int iret;
    char str[1024];

    AquesTalk_SetDevKey("xxxxxx"); // 開発ライセンスキーを指定
    AquesTalk_SetUsrKey("yyyyyy"); // 使用ライセンスキーを指定

    // 音声記号列を入力
    if(fgets(str, 1024-1, stdin)==0) return 0;

    // 音声合成
    // unsigned char *wav = AquesTalk_Synthe(str, 100, &size); //SJIS
    unsigned char *wav = AquesTalk_Synthe_Utf8(str, 100, &size);
    // unsigned char *wav = AquesTalk_Synthe_Utf16(wstr, 100, &size); // unsigned short *wstr
    if(wav==0){
        fprintf(stderr, "ERR:%d\n",size);
        return -1;
    }

    // 音声データ(wav フォーマット)の出力
    fwrite(wav, 1, size, stdout);

    // 音声データバッファの開放
    AquesTalk_FreeWave(wav);
}
```

```
return 0;
}
```

サンプルコードのビルド方法を次に示します。

```
$ g++ -o SampleTalk SampleTalk.c -L./lib64/f1 -lAquesTalk
```

サンプルコードの実行方法を次に示します。

```
$ echo "これわ、ごーせー/おんせーです。"|LD_LIBRARY_PATH=./lib64/f1 ./SampleTalk > test.wav
```

音声記号列にシェルの特許文字が含まれているので、ダブルクォーテーション(")で囲むなどエスケープする必要があります。

声種を変更する場合は、f1 の部分を他の声種に変更してください。

デフォルトの文字コードは UTF8 になっています。文字コードを変更する場合は、SampleTalk.c の AquesTalk\_Synthe\_Utf80 部分を文字コードに応じて変更してください。

test.wav に音声データが生成できれば OK です。エラーの場合には、文字コードやライブラリの配置が正しく行われているか再確認してください。

## アプリ開発ガイドライン

アプリケーションの開発(評価での使用を除く)は、以下のガイドラインに従ってください。

### ライセンスキー

本ライブラリの動作は、開発ライセンスキーと使用ライセンスキー、頒布ライセンスキーの3種類の関連キーに依存します。これらのキーは、各ライセンス購入時に発行されるライセンス証に記載されています。

AquesTalk\_SetDevKey () をアプリケーションの起動初期に一度呼び出します。引数には開発ライセンスキーを指定します。これにより製品版として動作し、評価版の制限がなくなります。

AquesTalk\_SetUsrKey()をアプリケーションの起動初期に一度呼び出します。AquesTalk\_SetDevKey()との呼び出し順序は任意です。引数には、使用ライセンスキー、または頒布ライセンスキーを指定します。この指定により、合成音声データに含まれる透かしが、使用ライセンス無しの状態から取得済みに変化します。この変化による聴感上の違いはありません。

頒布ライセンスによりアプリを配布する場合は、頒布ライセンスキーを指定して呼び出します。

それ以外の場合は、エンドユーザが使用ライセンスキーを指定できるようにします。なお、エンドユーザが個人かつ非営利の利用の場合は使用ライセンスが不要なので、使用ライセンスキーが未指定の場合は、この関数の呼び出しをスキップして構いません。

関数の戻り値のチェックは必ず行い、エラーの場合はエンドユーザにその旨を通知してください。

## 文書履歴

日付	版	更新内容
2006/05/17	1.0	新規作成
2006/10/06	1.1	C++のクラスからCの関数に変更
2011/02/05	1.3	Ver. 1.3 用に一部改訂
2020/11/20	1.7	X64 追加、多声種版
2020/11/20	1.7a	ライブラリ製品名を AquesTalk1 に修正
2025/04/04	2.0	Ver. 2.0 用に関数 IF、サンプル等の記載を全面改訂